

# INTRODUCTION TO DEVOPS

# Outline

- ▶ Intro to DevOps
- ▶ What is DevOps
- ▶ DevOps Lifecycle
- ▶ CI/CD Process
- ▶ Agile vs DevOps
- ▶ Continuous Development
- ▶ Continuous Testing
- ▶ Continuous Integration
- ▶ Continuous Deployment
- ▶ Continuous Monitoring
- ▶ DevOps Benefits
- ▶ Best Practices

# DEVOPS OVERVIEW



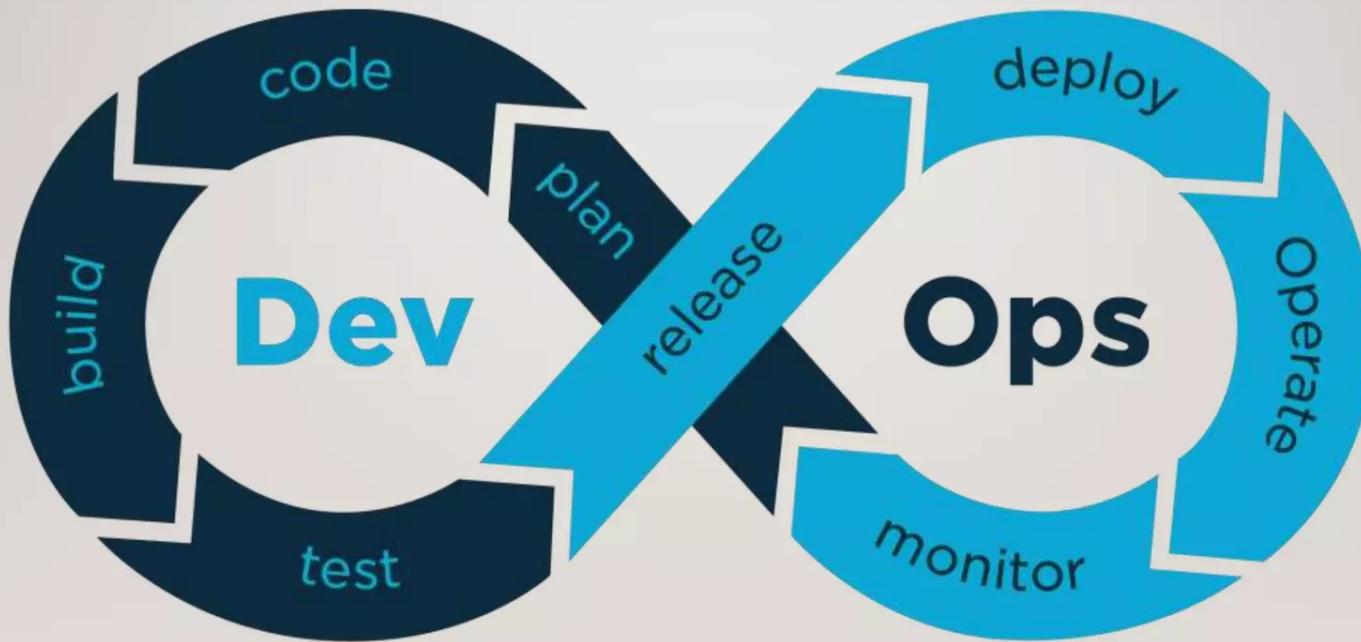
# What is DevOps ?

DevOps in simplest terms is “how you build and deploy your software”.

We all know, the word DevOps is a shortened acronym of Development and Operations. In essence, it is bringing together your software development team and operations team to work towards a common goal and have a unified vision. Their collective focus is always the product or the application which is being developed, rather than individual performances or targets.



DevOps is a software development methodology which improves the collaboration between developers and operations team using various automation tools. These automation tools are implemented using various stages which are a part of the DevOps Lifecycle



# Decoding DevOps



The assured benefits of DevOps philosophy are rapid software releases, shorter development cycles, reduced risks, quicker issue resolution, and better productivity. DevOps is a logical extension to agile and the main force behind the dynamism exhibited by many tech-giants for their extraordinary high performance. Shared responsibility, agile processes, automated tasks, smarter workflows, fearless innovation and continuous feedback are the driving factors of a **DevOps**.

# What Led To DevOps?

As we all know , there were many challenges with the conventional waterfall process. Software Development and IT Operations were two separate units that worked independently. Developers coded their piece of work in silos with unique configurations / environment. With manual internal processes and workflows, productivity is always hampered by process bottlenecks or other dependencies. All such error , short comings , issues were answered using DevOps.



# DevOps Principles

**Culture** => People, Process, Tools

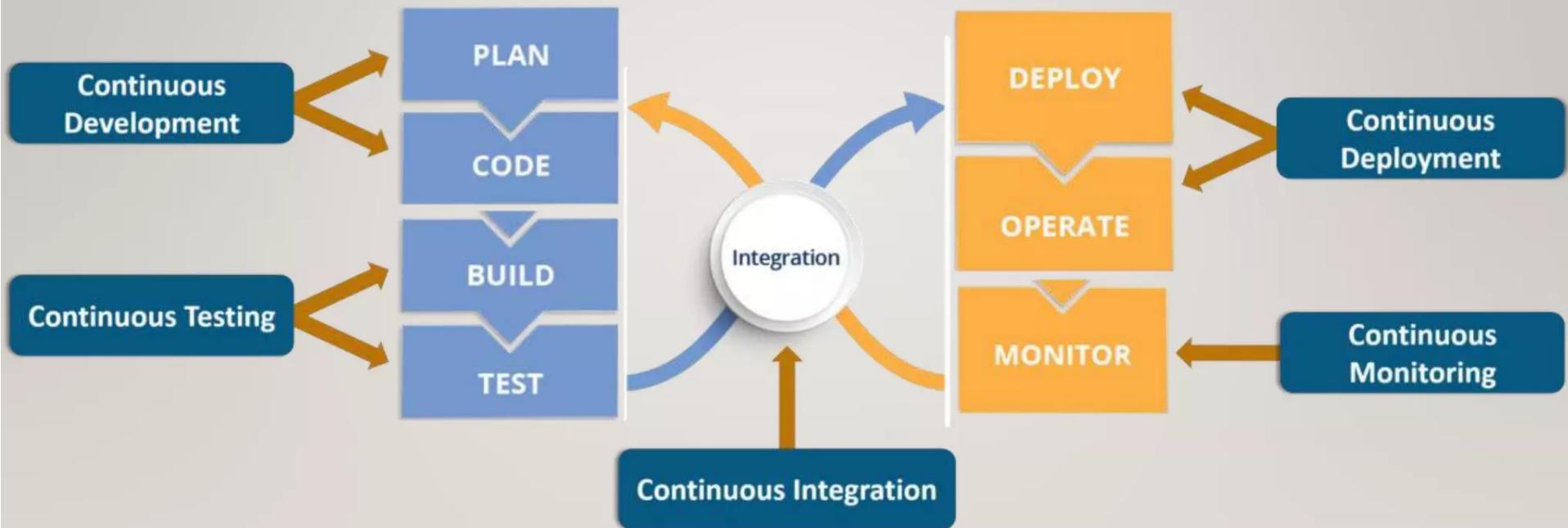
**Automation** => Infrastructure as Code

**Measurement** => Measure everything

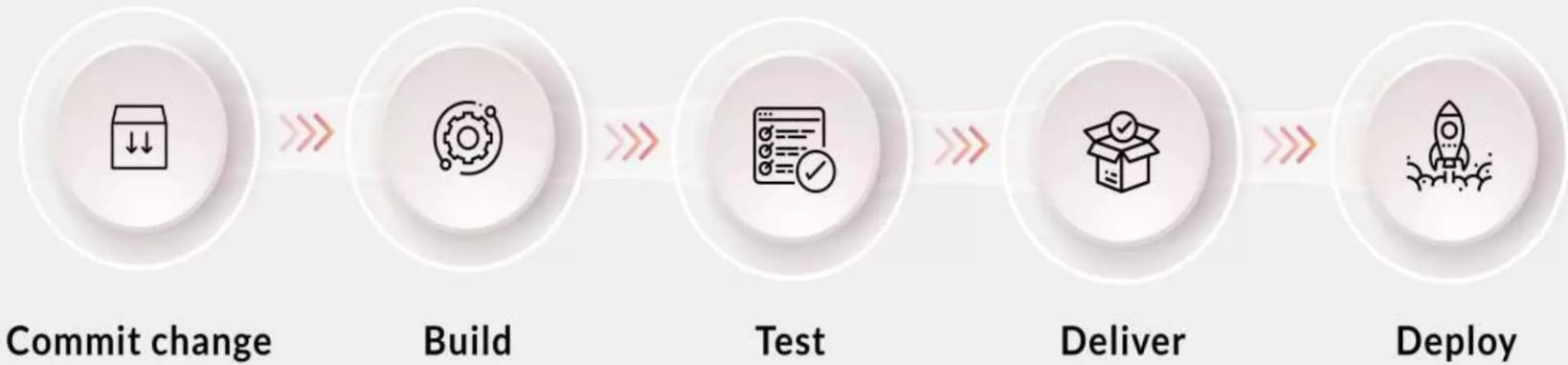
**Sharing** => Collaboration/Feedback

# DEVOPS LIFECYCLE

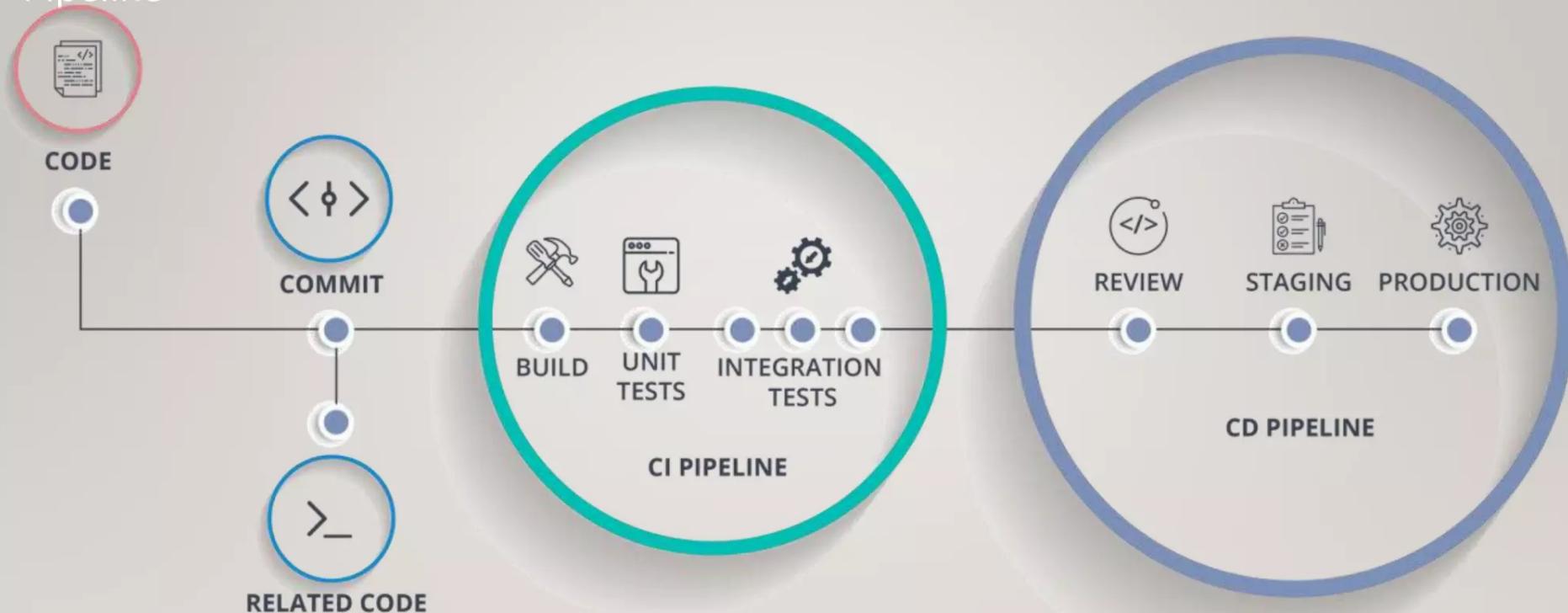
The DevOps Lifecycle divides the SDLC lifecycle into the following stages:



# CI/CD Process



# Automated CI/CD Pipeline



# The Important Phases In A DevOps Cycle



- Continuous Development
- Continuous Testing
- Continuous Integration
- Continuous Deployment
- Continuous Monitoring
- Continuous Feedback

# Continuous Development Phase

## Continuous Development

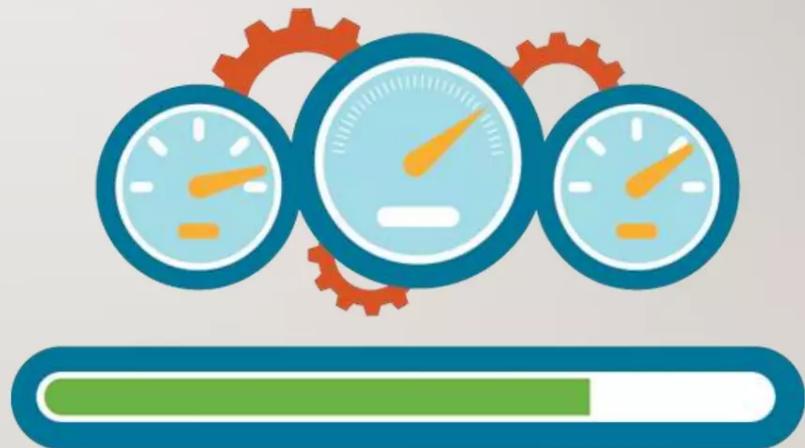
## Continuous Integration

## Continuous Deployment

## Continuous Testing

## Continuous Monitoring

This stage involves committing code to version control tools such as Git or SVN for maintaining the different versions of the code, and tools like Ant, Maven, Gradle for building/ packaging the code into an executable file that can be forwarded to the QAs for testing.



# Continuous Integration Phase

Continuous Development

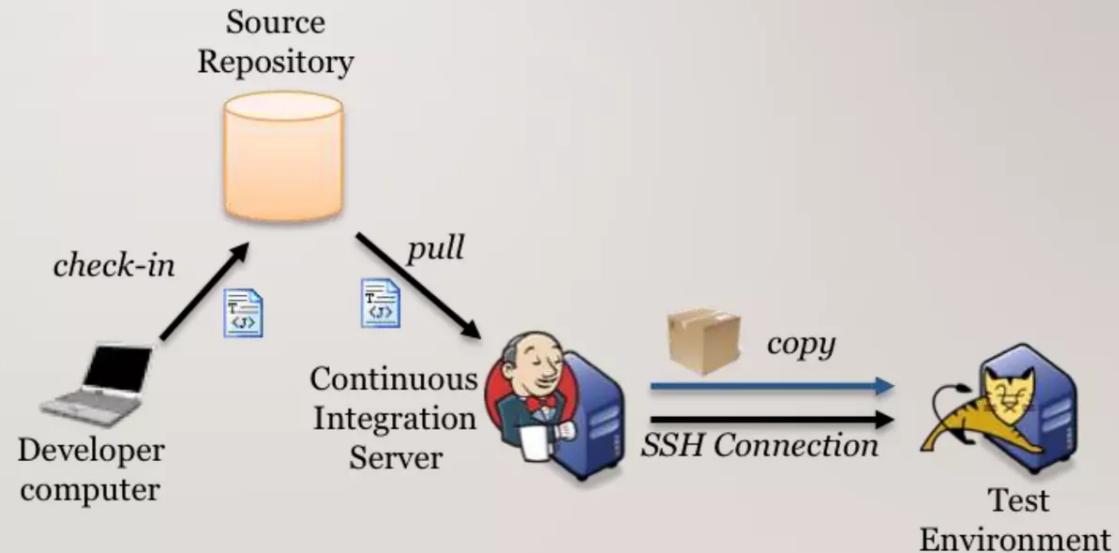
Continuous Integration

Continuous Deployment

Continuous Testing

Continuous Monitoring

The stage is a critical point in the whole DevOps Lifecycle. It deals with integrating the different stages of the DevOps lifecycle, and is therefore the key in automating the whole DevOps Process



# Continuous Deployment Phase

Continuous Development

Continuous Integration

Continuous Deployment

Continuous Testing

Continuous Monitoring

In this stage the code is built, the environment or the application is containerized and is pushed on to the desired server. The key processes in this stage are Configuration Management, Virtualization and Containerization



# Continuous Testing Phase

**Continuous Development**

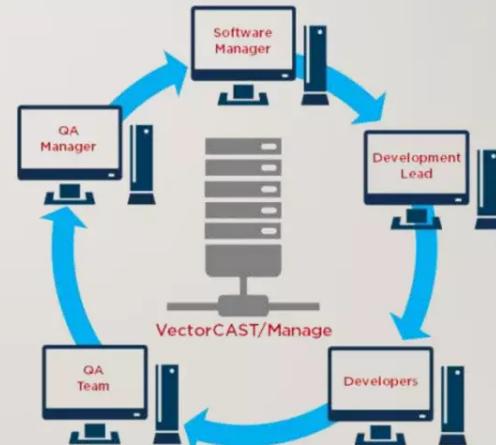
**Continuous Integration**

**Continuous Deployment**

**Continuous Testing**

**Continuous Monitoring**

The stage deals with automated testing of the application pushed by the developer. If there is an error, the message is sent back to the integration tool, this tool in turn notifies the developer of the error. If the test was a success, the message is sent to Integration tool which pushes the build on the production server



# Continuous Monitoring Phase

**Continuous Development**

**Continuous Integration**

**Continuous Deployment**

**Continuous Testing**

**Continuous Monitoring**

The stage continuously monitors the deployed application for bugs or crashes. It can also be setup to collect user feedback. The collected data is then sent to the developers to improve the application



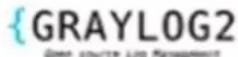
## Develop



Version everything.



## Log



## loggly



## Test



Jenkins



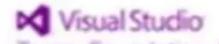
## Deploy



Capistrano



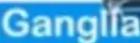
Jenkins



Visual Studio  
Team Foundation Server

## Monitor

**Nagios®**



pagerduty



Lantana

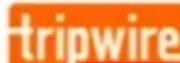
## Configuration Management



kubernetes



## Security



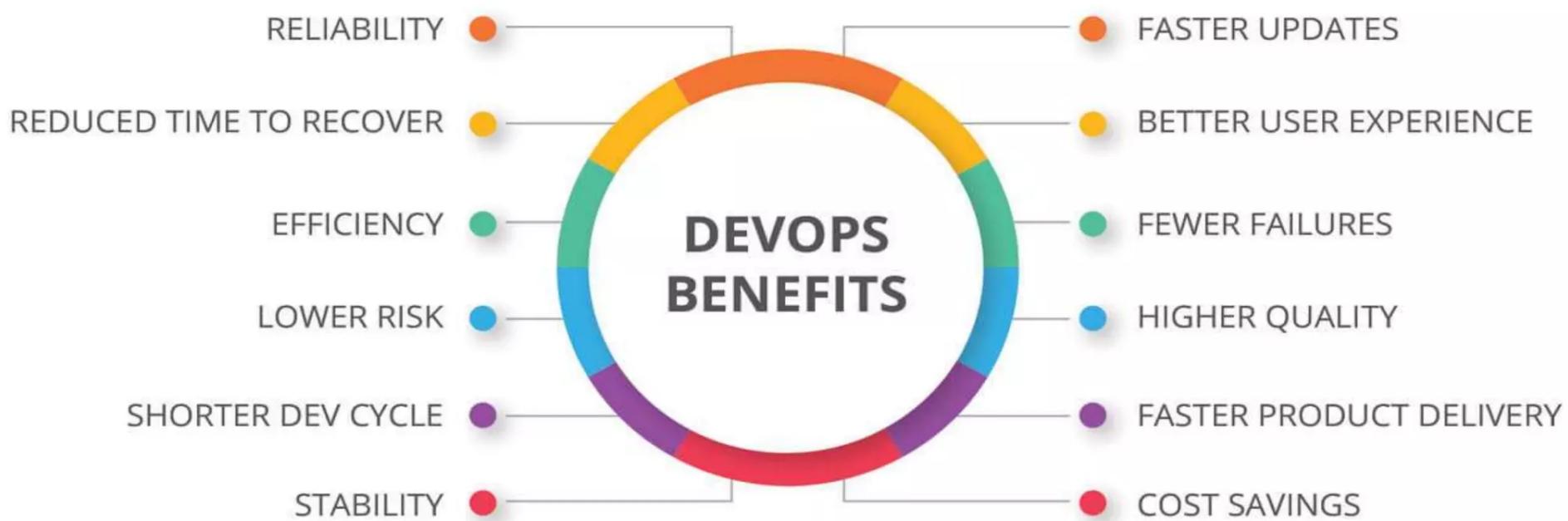
## Collaboration Platform



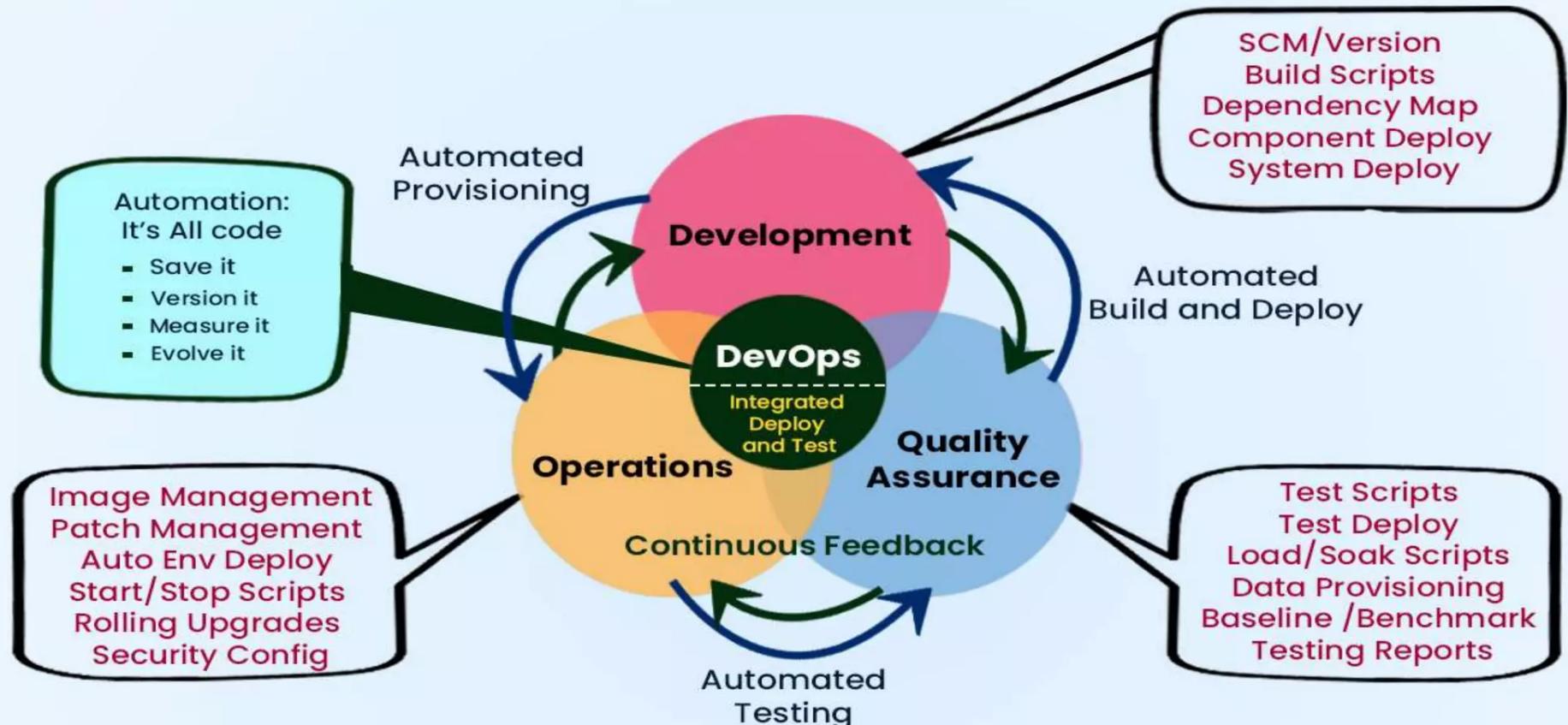
Visual Studio  
Team Foundation Server



# DEVOPS BENEFITS



# DevOps Automation Best Practices



# Thank You!